Inhoud

[**1.** De invoer van gegevens 2](#_Toc48653980)

[1.1 Inleiding 2](#_Toc48653981)

[1.2 INSERT INTO 2](#_Toc48653982)

[1.3 Verschillende foutmeldingen 4](#_Toc48653983)

# De invoer van gegevens

## Inleiding

Voor het **invoeren van gegevens** kan het DML-statement **INSERT INTO** worden gebruikt. Om dit statement correct te kunnen gebruiken, moet je wel antwoord weten op de onderstaande vragen.

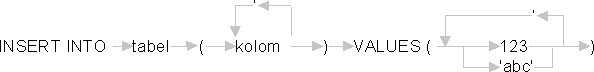
* + - In welke tabel moeten de gegevens worden ingevoerd?
    - Welke velden in die tabel moeten worden gevuld?
    - Met welke waarden moeten die velden worden gevuld?

Na bestudering van dit hoofdstuk dient de gebruiker tot het volgende in staat te zijn.

* + - Het kunnen toevoegen van records aan een tabel.
    - Verschillende foutmeldingen bij INSERT INTO kunnen begrijpen.

## INSERT INTO

Hieronder is het syntaxdiagram van het statement INSERT INTO afgebeeld.



Het statement INSERT INTO heeft de volgende syntax.

*INSERT INTO <tabelnaam> (<veld1>, <veld2>, …, <veldN>) VALUES (<waarde1>, <waarde2>, …, <waardeN>);*

Je gaat een klant toevoegen in de tabel tblKlanten in de KLANTEN database.

Hiervoor moet je eerst zorgen dat je deze database gaat gebruiken.

 Open in MySQL Workbench een nieuwe Query Tab en zorg dat je de database KLANTEN gaat gebruiken door het volgende statenent uit te voeren:

***use klanten;***

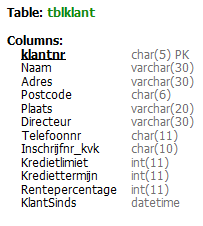
Daarna maak je gebruik van de drie vragen uit de inleiding.

#### Vraag 1 In welke tabel moeten de gegevens worden ingevoerd?

Alle klanten staan in de tabel *tblKlant*.

#### Vraag 2 Welke velden moeten worden gevuld?

Om dit te weten moet je eerst de structuur van de tabel kennen. Hieronder staat de structuur afgebeeld.

****

#### Vraag 3 Welke waarden moeten in deze velden staan?

Je voert de onderstaande gegevens in.

|  |  |
| --- | --- |
| ***Veld*** | ***Waarde*** |
| **Klantnr** | Roca |
| **Naam** | Roos international |
| **Adres** | Pannenschuurlaan 23 |
| **Postcode** | 5061 WE |
| **Plaats** | Oisterwijk |
| **Kredietlimiet**  **Klant sinds** | 10.000  10-9-2009 |

 Neem de onderstaande query over om de klant toe te voegen.

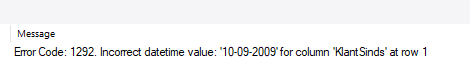
***INSERT INTO tblKlant (klantnr, naam, adres, postcode, plaats, kredietlimiet,KlantSinds)***

***VALUES ('Roca','Ross internationaal','Pannenschuurlaan 23','5061WE',*** ***'Oisterwijk', 1000,'2009-09-10');***

In deze query is duidelijk te zien dat er onderscheid gemaakt wordt tussen de verschillende gegevenstypen. Gegevens van het teksttype (o.a. Varchar en Char) en data worden tussen enkele aanhalingstekens geplaatst, bij numerieke gegevens gebeurt dat niet.

Is de waarde van een veld (nog) niet bekend, dan kan de waarde NULL worden gebruikt. Velden die niet in het statement worden opgenomen, worden automatisch met de waarde NULL gevuld.

Als je de onderstaande foutmelding krijgt, is het datumformaat niet juist.



De volgorde van de datum is namelijk afhankelijk van de database systeeminstellingen. De volgende varianten zijn mogelijk.

dd-mm-jjjj 10-9-2009

jjjj-mm-dd 2009-09-10 (komt het meest voor)

#### Om nu te kijken welke informatie in de tabel zit, kun je het volgende statement uitvoeren. We gaan hier in een volgende les verder op in.

 Open een nieuw Query Tab en bekijk de inhoud van de tabel tblKlant

***select \* from tblKlant;***

## Verschillende foutmeldingen

 Voeg nu een nieuwe klant toe met het volgende statement:

***INSERT INTO tblKlant (klantnr, naam, adres, postcode, plaats, kredietlimiet,KlantSinds)***

***VALUES ('SLI','SL internationaal','Straatweg 3',*** ***'Het Dorp', 1500,'2019-09-10');***

Je krijgt de volgende fout, omdat je 6 kolommen toevoegt en de tabel tblKlant heeft 7 kolommen (de postcode is vergeten):



 Voeg nu dezelfde klant nog een toe maar, nu met een postcode erbij. Gebruik het volgende statement:

***INSERT INTO tblKlant (klantnr, naam, adres, postcode, plaats, kredietlimiet,KlantSinds)***

***VALUES***

***('SLI','SL internationaal','Straatweg 3',*** ***'1234AA', 'Het Dorp', 1500, '2019-09-10');***

 Voeg nu een klant nog een toe maar, maar zet geen ‘ om de datum heen. Gebruik het volgende statement:

***INSERT INTO tblKlant (klantnr, naam, adres, postcode, plaats, kredietlimiet,KlantSinds)***

***VALUES***

***('GLI','GL internationaal','Lindelaan 7',*** ***'1234AB', 'Het Dorp', 1500,2019-09-10);***

Je krijgt de volgende fout, omdat de database nu ziet dat het een datum is en er mee gaat rekenen. 2019-09-10=2000 en dat is geen datum:



 Klant GLI woont niet op de Lindelaan, maar op de Beukenlaan. Je wil dit aanpassen door de klant nog een keer toe te voegen met het aangepaste adres:

***INSERT INTO tblKlant (klantnr, naam, adres, postcode, plaats, kredietlimiet,KlantSinds)***

***VALUES***

***('GLI','GL internationaal','Beukenlaan 7',*** ***'1234AB', 'Het Dorp', 1500, '2019-09-10');***

Dit gaat fout, want je het klantnr is een primary key, dus moet uniek zijn:



Als je deze wijziging van adres wil uitvoeren moet je een UPDATE statement uitvoeren of eerst dit record verwijderen. Hoe dat moet, wordt in een latere les behandeld.

 Voeg nu een klant met het volgende statement:

***INSERT INTO tblKlant (klantnr, naam, adres, postcode, plaats, kredietlimiet,KlantSinds)***

***VALUES***

***('ZZ','Zero Zero','Computerweg44', '3385AB', 500, 'Amersfoort', '2020-08-10');***

Je krijgt de volgende fout, omdat de waarde voor de *Kredietlimiet* en de *Plaats* omgedraaid is. Je probeert nu een stukje tekst in een kolom te stoppen, waar alleen cijfers (int(11)) in mogen:

